IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for executing an application ~~requested~~ comprising components in a client-server environment, comprising:

sending a request for the component from a client to a server, wherein the component ~~corresponds to~~ comprises a component script on the server, and wherein the client and the server have same functional capability;

transmitting component script including parameter information associated with the requested component of the script by the server to the client, wherein the parameter information is platform independent information; and

linking the transmitted component script including parameter information to a corresponding predefined structure ~~by~~ at the client to create an executable ~~parameter~~ script specific predefined structure of the requested component, wherein the predefined structure having an ~~intended~~ functionality corresponding to the intended functionality of the requested component, wherein the linking step comprises locating identifiers within the parameter information and inserting the component script data corresponding to the identifiers into the predefined structure; and

executing the script specific predefined structure of the component at the client.

2. (Canceled)

3. (Previously Presented) The executing method of claim 1, further comprising searching for the requested component in the script at the server in response to the request for the component from the client.

4. (Canceled) ~~The executing method of claim 1, wherein the linking step further comprises locating identifiers within the parameter information and inserting script data corresponding to the identifiers into the predefined structure.~~

5. (Previously Presented) The executing method of Claim 1, further comprising determining an access level of a user of the client, wherein the transmitting step further comprises transmitting the parameter information based on the access level of the user.

6. (Original) The executing method of claim 1, further comprising storing the predefined structure at the client and storing a copy of the predefined structure at the server so that there is a client predefined structure and a server predefined structure.

7. (Previously Presented) The executing method of claim 1, further comprising automatically deleting the parameter specific predefined structure after the user has exited the component.

8. (Previously Presented) The executing method of claim 1, further comprising establishing a connection in response to the request for the component from the client, and the server creating a session identification number for the connection so that the client and the server can follow a connectionless protocol.

9. (Previously Presented) A system for executing a component, comprising:

a client including a client memory, and a client processor, a client run time engine configured to reside in the client memory, wherein the client run time engine comprises a plurality of client predefined structures; and

a server including a server memory and a server processor, a server run time engine configured to reside in the server memory, wherein the server and the client have same functional capability with respect to the client run time engine and the server run time engine, and wherein the client run time engine sends a request for a component to the server, wherein the component corresponds to a script on the server, and wherein the server run time engine transmits parameter information associated with the requested component of the script to the client, and wherein the client run time engine links the parameter information received from the server with a corresponding client predefined structure of the plurality of client predefined structures to create an executable parameter specific predefined structure, and wherein the

predefined structure having an intended functionality corresponding to the intended functionality of the requested component.

10. (Canceled)

11. (Canceled)

12. (Previously Presented) The system of claim 9, wherein the client run time engine comprises a client parser and a client execution engine, wherein the client execution engine comprises a client linker and the plurality of client predefined structures, wherein the client parser configured to instruct the client processor to search for identifiers within the parameter information transmitted by the server, wherein the client linker configured to instruct the client processor to link the parameter information to the client predefined structure to create the executable parameter specific predefined structure.

13. (Previously Presented) The system of claim 12, wherein the server run time engine comprises a server parser and a server execution engine, the server execution engine comprises a server linker and a plurality of server predefined structures, a server predefined structure having an intended functionality corresponding to an intended functionality of a component type of a plurality of component types, wherein the component has the intended functionality of the component type, the server parser configured to instruct the server processor to search for the component in the script, the component being requested by the client, the server linker configured to instruct the server processor to link the parameter information to a corresponding server predefined structure to provide a specific predefined structure; and a server transceiver being configured to transmit the parameter information associated with the component of the script.

14. (Previously Presented) The system of claim 9, wherein client memory further comprises a client long term memory and a client short term memory, the client run time engine being stored in the client long term memory before the client sends the request for the component of the

script, wherein the client processor is configured to transfer the client run time engine to the client short term memory when the client sends the request for the component of the script, to temporarily store the executable parameter specific predefined structure in the client short term memory, and to automatically delete the executable parameter specific predefined structure from the client short term memory when the client exits the component.

15. (Previously Presented) The system of claim 9, wherein the parameter information transmitted by the server includes identifiers associated with component information and the predefined structure includes the corresponding identifiers.

16. (Original) The system of claim 9, wherein the server creates a unique session identification number for every connection established to uniquely identify each connection and recreate the session previously established thereby facilitating a connectionless protocol.

17. (Previously Presented) An application for executing a component when a user accesses a component on a system, the application comprising:

a first run time engine comprising an execution engine comprising a plurality of predefined structures and a linker, a predefined structure of the plurality of predefined structures having an intended functionality of a component type of a plurality of component types, wherein the component has the intended functionality of the component types, wherein the component corresponds to a script on a server, and wherein, when the user accesses the component:

(a) the linker instructs a client processor to link parameter information associated with the component to a corresponding predefined structure to create an executable parameter specific predefined structure, the parameter information associated with the component being transmitted from the server to a client and stored in a client processor readable memory, wherein the server and the client have the same functional capability; and

(b) the execution engine instructs a client processor to execute the executable parameter specific predefined structure to execute the component; wherein the first run time engine is stored in a media and the first run time engine is transferred to the client

processor readable memory of a system including the client processor readable memory and the client processor when the media is used with the system.

18. (Previously Presented) The application of claim 17, wherein the system comprises a server, the server comprising a server processor readable memory, a server transceiver, a server processor and a server run time engine, wherein the server run time engine is transferred to a server processor readable memory of the system and the server run time engine comprises a copy of the first run time engine, wherein the server run time engine comprises a server parser and a server execution engine, wherein a user at the client requests a component from the server prior to running the component and, when the user requests the component:

(a) the server parser instructs the server processor to search for the component in the script, the script being stored in the server processor readable memory, and

(b) the execution engine instructs the server processor to transmit the parameter information associated with the component of the script to the client via the server transceiver.

19. (Previously Presented) The application of claim 18, wherein the server execution engine further comprises a plurality of server predefined structures, a server predefined structure of the plurality of server predefined structures having the intended functionality of a component type of the plurality of component types, wherein the component requested by the user has the intended functionality of the component type.

20. (Original) The application of claim 19, wherein, when the client requests the component, the server execution engine instructs the server processor to create a session number and to transmit the session number to the client.

21. (Previously Presented) The application of claim 17, wherein the execution engine instructs the client processor to store the executable parameter specific predefined structure in the client processor readable memory and instructs the processor to automatically delete the

executable parameter specific predefined structure from the memory after the user exits the component.

22. (Canceled)

23. (Canceled)

24. (Canceled)

25. (Canceled)

26. (Canceled)

27. (Canceled)

28. (Canceled)

29. (Canceled)

30. (Canceled)